ซอฟต์แวร์

สุทธิศักดิ์ พงศ์ธนาพาณิช

เปิดโลกมัลติมีเดียด้วย VB/Win ตอน โปรแกรม MIDI File Player

แนะนำวิธีการเขียนโปรแกรม โดยใช้ฟังก์ชันวินโดว์ API ด้านมัลติมีเดีย

บทความเปิดโลกมัลติมีเดียตอนที่ผ่าน ๆ มา ผู้เขียนมักจะเน้นหนักไปที่ด้านของไฟล์ .wav เป็นส่วนใหญ่ แต่ก็มีผู้อ่านบางท่านจดหมายมายังตู้ ป.ณ. 35 โดยต้องการให้เพิ่มเติมบทความด้านของไฟล์ MIDI กันบ้าง ดังนั้นในบทความตอนนี้ผู้เขียนก็จะแนะนำวิธีการเขียนโปรแกรมโดยใช้ฟังก์ชันวินโดว์ API ด้านมัลติมีเดีย เพื่อจัดการกับไฟล์ MIDI โดยในบทความตอนนี้ผู้เขียนก็ได้เขียนโปรแกรมที่ค่อนข้างจะสมบูรณ์เพื่อให้ผู้อ่านสามารถนำไปใช้หรือพัฒนาต่อได้ทันที โดยโปรแกรมประจำบทความตอนนี้ ก็ยังคงรูปแบบการออกแบบที่ใกล้เคียงกับโปรแกรม CDAUDIO ในบทความตอนที่แล้ว ทั้งนี้ก็เพื่อให้ผู้อ่านมีความคุ้นเคยกับการเขียนโค้ดของโปรแกรม และสามารถติดตามการทำงานของโปรแกรมได้ง่าย และยังเป็นการช่วยลดเวลาสำหรับการเรียนรู้อีกด้วย

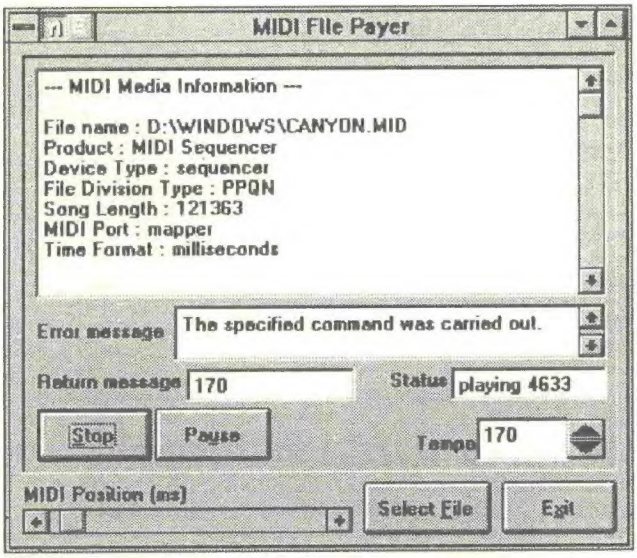
เนื่องจากในปัจจุบันก็มีนักดนตรีอยู่ส่วนหนึ่งที่ใช้คอมพิวเตอร์ในการสร้างทำนองดนตรี ซึ่งไฟล์ที่ได้ก็มักจะเป็นไฟล์ในรูปแบบ MIDI ดังนั้นผู้เขียนจึงเห็นว่าการเรียนรู้ถึงโครงสร้างของไฟล์ MIDI และวิธีการจัดการข้อมูลแต่ละไบต์ของไฟล์ MIDI นั้นมีความสำคัญเป็นอย่างมาก เพราะจะเป็นพื้นฐานที่โปรแกรมเมอร์ ในส่วนของคนทำดนตรีสามารถที่จะนำความรู้ในส่วนนี้ไปพัฒนาต่อเป็นรูปแบบของเพลงคาราโอเกะ (มีเนื้อเพลงประกอบดนตรี) ได้อย่างสมบูรณ์กว่าเทคนิคที่กำลังใช้ ๆ กันอยู่ในปัจจุบัน ดังนั้นสำหรับผู้อ่านที่ต้องการที่จะเรียนรู้ถึงฟังก์ชันวินโดว์ API ด้านมัลติมีเดียระดับล่างสำหรับการจัดการกับไฟล์ MIDI นั้น ก็ขอให้อดใจรออีกสักระยะผู้เขียนก็จะนำบทความเหล่านี้มาเสนอในตอนต่อ ๆ ไป

โปรแกรม MIDI File Player

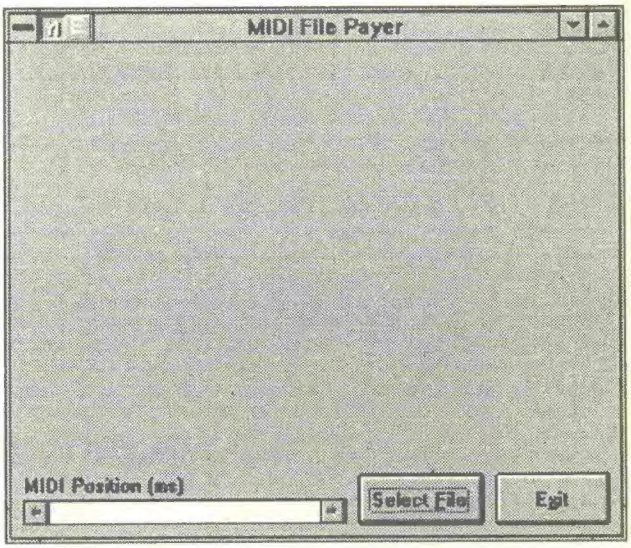
โปรแกรม MIDI File Player เป็นโปรแกรมสำหรับการเล่นกลับ (playback) ไฟล์ MIDI (.MID) โดยที่ในขณะที่กำลังเล่นกลับไฟล์ MIDI นั้นสถานะการเล่นกลับก็จะถูกแสดงผลในช่อง Status และตำแหน่งการเล่นกลับปัจจุบันก็จะแสดงบนลงในแถบ scroll bar ด้านล่างของโปรแกรม ดังตัวอย่างในรูปที่ 1 แสดงว่าในปัจจุบันไดรเวอร์สำหรับการเล่นกลับไฟล์ MIDI กำลังเล่นกลับดนตรีที่ตำแหน่ง 4633 (playing 4633) เป็นต้น

โดยในระหว่างการเล่นกลับนั้นผู้อ่านยังสามารถควบคุมความเร็วในการเล่นกลับ (tempo) ซึ่งก็หมายถึงการควบคุมจังหวะของดนตรีให้เร็วขึ้นหรือลดลงได้โดยการใส่ตัวเลขจำนวนเต็มลงในช่อง Tempo แล้วกดคีย์ Enter เท่านั้น ผู้อ่านก็จะสามารถควบคุมการเล่นกลับได้ตามที่ต้องการ

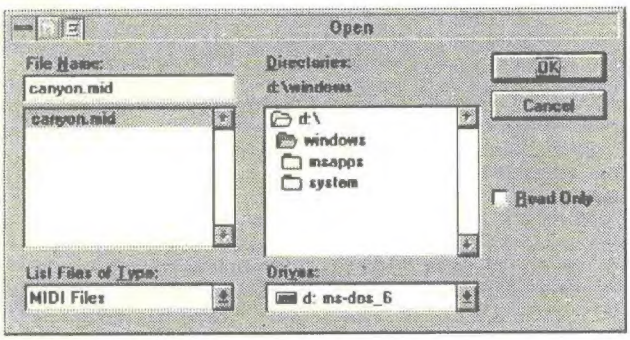
ที่สำคัญโปรแกรมนี้ยังอนุญาตให้ผู้อ่านสามารถควบคุมตำแหน่งการเล่นของแต่ละเพลงแบบ real-time ได้อีกด้วย เช่น สมมติว่าในปัจจุบันผู้อ่านกำลังเล่นเพลง



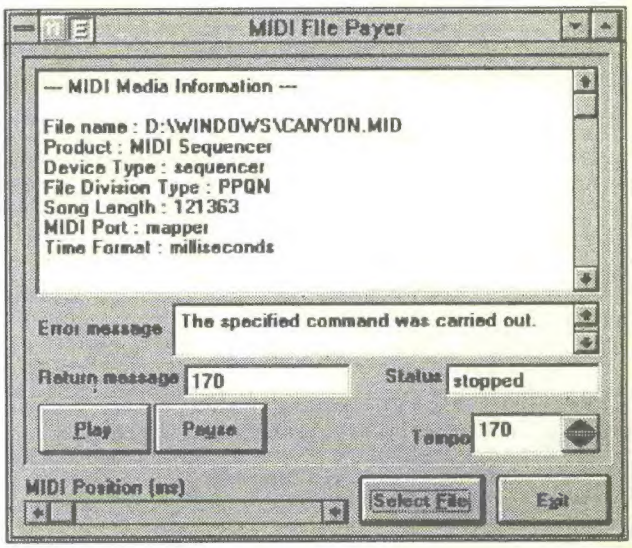
รูปที่ 1 แสดงฟอร์ม Form1 ของโปรแกรม MIDIDEMO



รูปที่ 2 แสดงฟอร์มเริ่มต้นของ MIDIDEMO



รูปที่ 3 แสดงการเลือกไฟล์ CANYON.MID
จากไดอะล็อกบ็อกซ์ Open



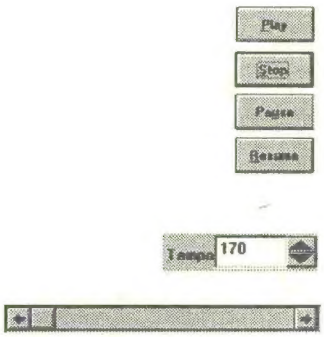
รูปที่ 4

ดังในรูปที่ 1 และต้องการกระโดดไปเล่นเพลงที่ตำแหน่ง 2 นาที 50 วินาที ก็สามารถกระทำได้โดยการใช้เม้าส์ลาก cursor ของ scroll bar มาทางขวา ก็ได้ โปรแกรมก็จะกระโดดไปเล่นเพลงยังตำแหน่งนั้น ๆ ทันที ซึ่งจะทำให้สะดวกในการควบคุมการเล่นกลับไฟล์ MIDI เป็นอย่างมาก

สำหรับการเริ่มใช้งานโปรแกรม MIDIDEMO นั้นก็สามารถกระทำได้โดยให้ผู้อ่านสั่งรันโปรแกรมภายใต้เอนไวรอนเมนท์ของ VB/Win จากนั้นโปรแกรมก็แสดงผล Form1 ดังในรูปที่ 2 จากนั้นผู้อ่านก็ต้องทำการเลือกไฟล์ MIDI ที่ต้องการโดยการคลิ๊กที่ปุ่มคำสั่ง Select File เพื่อแสดงผลไดอะล็อกบ็อกซ์ Open และผู้อ่านก็สามารถเลือกไฟล์ MIDI ที่ต้องการได้ ดังตัวอย่างในรูปที่ 3 แสดงถึงการเลือกไฟล์ที่ชื่อ CANYON.MID ซึ่งเป็นไฟล์ตัวอย่างที่มากับวินโดว์ และอยู่ในไดเรคทอรี c:\windows

เมื่อผู้อ่านเลือกไฟล์ MIDI ที่ต้องการแล้ว โปรแกรมที่จะทำการเปิดไฟล์ดังกล่าวในโดยอัตโนมัติ และก็จะทำการอ่านรายละเอียดของไฟล์ MIDI และรายงานผลที่ได้ตัวอย่างในรูปที่ 4

เมื่อไฟล์ถูกเลือกแล้วผู้อ่านก็สามารถที่จะคลิ๊กที่ปุ่มคำสั่งต่าง ๆ เพื่อควบคุมการเล่นกลับดนตรีได้ดังนี้



ปุ่มคำสั่งสำหรับการเล่นกลับดนตรี

ปุ่มคำสั่งสำหรับการหยุดการเล่นกลับดนตรี

ปุ่มคำสั่งสำหรับการหยุดการเล่นกลับดนตรีชั่วคราว

ปุ่มคำสั่งสำหรับการการเล่นกลับดนตรีต่อจากตำแหน่งที่ถูกสั่งหยุดด้วยคำสั่ง Pause

ช่องสำหรับกรอกตัวเลขจำนวนเต็ม เพื่อกำหนดความเร็วของจังหวะดนตรี

แถบควบคุมตำแหน่งการเล่นของแต่ละเพลง

การทำงานของโปรแกรม MIDI File Player

เมื่อโปรแกรมเริ่มต้นทำงาน รูทีน Form_Load ก็จะถูกเรียกใช้เป็นรูทีนแรกสุดเมื่อมีการโหลดฟอร์ม Form1 มายังหน่วยความจำ จากนั้นก็กำหนดให้คอนโทรล Text1 ถึง Text4 เป็น textbox ชนิดที่ไม่สามารถแก้ไขข้อความได้ (read only) โดยการใช้ฟังก์ชันวินโดว์ API SendMessage และเขียนโค้ดเพื่ออ่านขนาดของสตริง mcilpstrReturnString และ lpstrReturnString ซึ่งเป็นสตริงสำหรับฟังก์ชัน API มาเก็บเอาไว้สำหรับการใช้งานต่อไป และสุดท้ายก็เป็นการย้ายฟอร์มมายังกึ่งกลางจอภาพ

```
Private Sub Form_Load()
    ' Set Text1 - Text4 to read only textbox.
    Dummy% = SendMessage(Text1.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text2.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text3.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text4.hWnd, EM_SETREADONLY, 1, ByVal 0&)


    mcilpstrReStrLen% = Len(mcilpstrReturnString) - 1
    lpstrReStrLen% = Len(lpstrReturnString) - 1

    ' Show form at middle of screen.
    Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height - Me.Height) / 2
End Sub
```

จากนั้นถ้าหากผู้อ่านต้องการเล่นกลับไฟล์ MIDI ผู้อ่านก็ต้องเลือกที่ปุ่มคำสั่ง Select File ซึ่งรูทีนเหตุการณ์ Command6_Click() ก็จะถูกเลือก โดยโค้ดภายในรูทีนเหตุการณ์นี้ก็จะเริ่มต้นด้วยโค้ดสำหรับการแสดงผลไดอะล็อกบ็อกซ์ Open เพื่อให้ผู้อ่านเลือกไฟล์ที่ต้องการ

```
On Error Resume Next

CMDialog1.filename = mciFilename$
CMDialog1.Filter = "MIDI Files|*.mid|All Files|*.*"
CMDialog1.Flags = cdlOFNFileMustExist
CMDialog1.ShowOpen

If Err = 32755 Then Exit Sub

mciFilename$ = CMDialog1.filename
```

เมื่อผู้อ่านเลือกไฟล์แล้ว ก็จะมีการตรวจสอบสถานะภาพของตัวแปร OpenFlag โดยถ้าหากมีค่าเท่ากับ True ก็แสดงว่ามีไฟล์ถูกเปิดก่อนหน้านี้แล้ว เราก็ต้องทำการปิดไฟล์ดังกล่าวเสียก่อนทุกครั้งดังนี้

```
' Close last opened media.
If OpenFlag Then
    mciErrCode& = mciSendString("close mciVBIS", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
             mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
End If
```

จากนั้นจึงทำการเปิดไฟล์ที่ถูกเลือกและกำหนดให้มีค่า device_id เท่ากับ mciVBIS ทั้งนี้ก็เพื่อให้เกิดความสะดวกในการอ้างอิงชื่อดีไวซ์สำหรับการใช้งานโดยคำสั่งอื่น ๆ ต่อไป และเมื่อเปิดไฟล์แล้วก็ทำการกำหนดให้หน่วยของเวลาที่ใช้ในการกำหนดจังหวะของการเล่นกลับดนตรีเท่ากับหน่วยมิลลิวินาที (millisecond) ซึ่งมีความละเอียดถึง 1/1,000 วินาที

```
    OpenFlag = True
    ' Open media.
    mciErrCode& = mciSendString("open " & mciFilename$ & _
                  " type sequencer alias mciVBIS", lpstrReturnString,
lpstrReStrLen%, 0)

    ' Get time format to millisec.
    mciErrCode& = mciSendString("set mciVBIS time format ms", _
                  lpstrReturnString, lpstrReStrLen%, 0)
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString,
mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
```

ส่วนถัดมาของรูทีนนี้ก็จะเป็นการอ่านรายละเอียดและคุณสมบัติของไฟล์ MIDI ที่ถูกเปิด เช่น ชื่อไฟล์ที่ถูกเปิด ความสามารถของดีไวซ์ ความยาวของดนตรี เป็นต้น เพื่อรายงานต่อไปดังนี้

```
' Get information of current media (opened file).
p$ = "- MIDI Media Information -" & vbCrLf & vbCrLf
mciErrCode& = mciSendString("info mciVBIS file", lpstrReturnString, _
              lpstrReStrLen%, 0)
p$ = p$ & "File name : " & Ztrim(lpstrReturnString) & vbCrLf
mciErrCode& = mciSendString("info mciVBIS product", lpstrReturnString, _
              lpstrReStrLen%, 0)
p$ = p$ & "Product : " & Ztrim(lpstrReturnString) & vbCrLf
mciErrCode& = mciSendString("capability mciVBIS device type", _
              lpstrReturnString, lpstrReStrLen%, 0)
p$ = p$ & "Device Type : " & Ztrim(lpstrReturnString) & vbCrLf
mciErrCode& = mciSendString("status mciVBIS division type", _
              lpstrReturnString, lpstrReStrLen%, 0)
p$ = p$ & "File Division Type : " & Ztrim(lpstrReturnString) & vbCrLf
mciErrCode& = mciSendString("status mciVBIS length", lpstrReturnString, _
                                                                    →
```

```
                lpstrReStrLen%, 0)
    mciMIDIlength& = Val(lpstrReturnString)
    p$ = p$ & "Song Length : " & mciMIDIlength& & vbCrLf
    mciErrCode& = mciSendString("status mciVBIS port", lpstrReturnString, _
                lpstrReStrLen%, 0)
    p$ = p$ & "MIDI Port : " & Ztrim(lpstrReturnString) & vbCrLf
    mciErrCode& = mciSendString("status mciVBIS time format", lpstrReturnString, _
                lpstrReStrLen%, 0)
    p$ = p$ & "Time Format : " & Ztrim(lpstrReturnString) & vbCrLf
    Text1.Text = p$
```

จากนั้นก็กำหนดความยาวของดนตรีให้กับ scroll bar และสุดท้ายก็เป็นการอ่านความเร็วของจังหวะของดนตรีเพื่อแสดงผลในช่อง Tempo ต่อไป

```
    ' Set wave length (ms) to HScroll.
    HScroll1.Min = 1
    HScroll1.Max = mciMIDIlength& / 1000

    ' Get default tempo.
    mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                lpstrReStrLen%, 0)
    mciTempo% = Val(lpstrReturnString)
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString,
mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    Text5.Text = mciTempo%
```

จากนั้นผู้อ่านก็สามารถที่จะเล่นกลับหรือหยุดการเล่นดนตรีได้โดยการคลิ๊กที่ปุ่มคำสั่ง Play หรือ Stop ตามลำดับ ซึ่งในการคลิ๊กปุ่มคำสั่งดังกล่าว ก็จะทำให้เกิดการเรียกรูทีนเหตุการณ์ Command1_Click() ซึ่งภายในรูทีนเหตุการณ์ดังกล่าวก็จะมีการเขียนโค้ดเพื่อเล่นกลับหรือหยุดการเล่นดนตรีดังนี้

```
    If Command1.Caption = "&Play" Then
        Command1.Caption = "&Stop"
        mciErrCode& = mciSendString("play mciVBIS from 1", _
                    lpstrReturnString, lpstrReStrLen%, 0)

        ' Call this procedure to set tempo.
        Text5_KeyPress 13
    Else
        Command1.Caption = "&Play"
        mciErrCode& = mciSendString("stop mciVBIS", _
                    lpstrReturnString, lpstrReStrLen%, 0)
    End If
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString,
mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
```

ซึ่งโค้ดจะประมวลผลส่วนของการเล่นกลับ (Play) หรือหยุดการเล่นกลับ (Stop) ก็ได้จากการเปรียบเทียบคุณสมบัติ Caption ของคอนโทรล Command1 ดังข้างต้น ส่วนในตอนท้ายก็จะมีการตรวจสอบความสำเร็จของการเรียกใช้คำสั่งโดยใช้ฟังก์ชัน mciGetErrorString

และในระหว่างการเล่นกลับนั้นผู้อ่านสามารถที่จะหยุดการเล่นกลับชั่วคราวได้โดยการคลิ๊กที่ปุ่มคำสั่ง Pause และเล่นต่อจากตำแหน่งที่หยุดได้ด้วยคำสั่ง Resume ซึ่งเมื่อคลิ๊กที่ปุ่มคำสั่งเหล่านี้ก็จะทำให้เกิดเหตุการณ์ Command2_Click() ทันที ซึ่งภายในรูทีนเหตุการณ์นี้ก็จะมีการเขียนโค้ดดังนี้

```
If Command2.Caption = "Pa&use" Then
    mciErrCode& = mciSendString("pause mciVBIS", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    Command2.Caption = "&Resume"
Else
    mciErrCode& = mciSendString("resume mciVBIS", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    Command2.Caption = "Pa&use"
End If
Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
```

แต่สำหรับคำสั่ง Resume นั้น sequencer บางตัวไม่สามารถปฏิบัติคำสั่งนี้ได้ เช่น MCISEQ ซึ่งเป็นไดรเวอร์มาตรฐานของไมโครซอฟท์เป็นต้น ดังนั้นถ้าหากผู้อ่านมีการใช้ sequencer ตัวอื่น ๆ นอกเหนือจากนี้ ผู้อ่านก็ต้องทำการทดสอบคำสั่งนี้เองจากการทดลองใช้งานโปรแกรม

สำหรับ sequencer ที่ไม่รับคำสั่ง Resume นี้ ผู้อ่านก็สามารถที่จะเขียนโค้ดเพื่อจำลองการทำงานของคำสั่ง Resume ได้โดยทำตามลำดับขั้นตอนดังนี้

1. เมื่อมีการคลิ๊กที่ปุ่มคำสั่ง Pause ให้มีการจัดเก็บตำแหน่งการเล่นกลับปัจจุบันไว้ในตัวแปร ชนิดเลขจำนวนเต็ม long

2. หยุดการเล่นกลับด้วยคำสั่ง Stop

3. เมื่อมีการคลิ๊กที่ปุ่มคำสั่ง Resume ให้มีการใช้คำสั่ง Seek เพื่อกำหนดตำแหน่งที่จะเริ่มต้นเล่นจากค่าของตัวแปรในขั้นตอนที่ 1 และตามด้วยคำสั่งPlay เพื่อเล่นกลับ (สมมติว่าต้องการเล่นกลับจากตำแหน่ง 1000) ดังนี้

```
mciErrCode& = mciSendString("seek mciVBIS to 1000" , _
          lpstrReturnString, lpstrReStrLen%, 0)
mciErrCode& = mciSendString("play mciVBIS ", lpstrReturnString, lpstrReStrLen%, 0)
```

ในขณะที่กำลังเล่นกลับดนตรีนั้น ผู้อ่านก็สามารถเปลี่ยนตำแหน่งการเล่นกลับไปข้างหน้าหรือถอยหลังได้ตามต้องการ โดยการลาก scroll bar cursor ของแถบ scroll bar ซึ่งจะทำให้เกิดการเรียกใช้รูทีน HScroll1_Scroll ซึ่งภายในรูทีนนี้ก็จะเริ่มต้นด้วยการหยุดการทำงานของ Timer ชั่วคราว จากนั้นก็เลื่อนพอยน์เตอร์สำหรับการเล่นกลับไปยังตำแหน่งที่กำหนดและสั่งเล่นกลับที่ตำแหน่งดังกล่าว

```
' Disable Timer before seek.
Timer1.Enabled = False

mciErrCode& = mciSendString("seek mciVBIS to " & 1000& * HScroll1.Value, _
        lpstrReturnString, lpstrReStrLen%, 0)
mciErrCode& = mciSendString("play mciVBIS ", lpstrReturnString, lpstrReStrLen%, 0)
```

และกำหนดความเร็วของจังหวะดนตรีอีกครั้ง และท้ายสุดก็เป็นการเริ่มต้นการทำงานของ Timer ตามปกติ

```
' Set tempo.
mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
              lpstrReturnString, lpstrReStrLen%, 0)
mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
              lpstrReStrLen%, 0)
mciTempo% = Val(lpstrReturnString)
Text5.Text = mciTempo%
Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString

' Enable Timer after play.
Timer1.Enabled = True
```

ส่วนความสามารถสุดท้ายของโปรแกรมสำหรับการเล่นกลับดนตรี MIDI ก็ได้แก่การกำหนดความเร็วของจังหวะดนตรี (Tempo) ซึ่งผู้อ่านสามารถกระทำได้ 2 วิธี ดังนี้

1. ใส่ค่าตัวเลขจำนวนเต็มลงในช่อง Tempo และกดคีย์ Enter

ซึ่งการกำหนดค่า Tempo ด้วยวิธีนี้ ก็จะถูกเขียนโค้ดลงในรูทีนเหตุการณ์ Text5_KeyPress ซึ่งจะมีการตรวจสอบคีย์ที่ถูกกด โดยจะยอมรับเฉพาะค่าตัวเลข 0 ถึง 9 เท่านั้น และเมื่อผู้อ่านกดคีย์ Enter ก็จะมีการตรวจสอบค่าที่ได้ โดยถ้าหากค่าที่ได้ถูกต้องก็จะมีการกำหนด Tempo ของดนตรีด้วยโค้ดบรรทัดที่แสดงด้วยตัวอักษรแบบหนาข้างล่าง

```
Select Case KeyAscii
Case 48 To 57: ' accept value.
Case 13
    tmp = Val(Text5.Text)
    If tmp <= 32767 Then
        mciTempo% = tmp
        mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                      lpstrReturnString, lpstrReStrLen%, 0)

        mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        mciTempo% = Val(lpstrReturnString)
        Text5.Text = mciTempo%
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
End Select
```

2. คลิ๊กที่ Spin button เพื่อเพิ่มหรือลดค่าตัวเลขในช่อง Tempo

โดยถ้าหากผู้อ่านคลิ๊กที่ปุ่มลูกศรขึ้น ค่าตัวเลขก็จะเพิ่มขึ้น ซึ่งก็จะมีการเรียกโค้ดในรูทีนเหตุการณ์ Spin1_SpinUp() แต่ถ้าหากผู้อ่านคลิ๊กที่ปุ่มลูกศรลง ค่าตัวเลขก็จะลดลง ซึ่งก็จะมีการเรียกโค้ดในรูทีนเหตุการณ์ Spin1_SpinDown() ซึ่งทั้งสองรูทีนเหตุการณ์นี้ มีโค้ดที่แตกต่างกันก็เฉพาะการตรวจสอบค่าตัวเลขเท่านั้นว่ามีค่าที่ถูกต้องหรือไม่จากนั้นก็เพิ่มหรือลดค่าของ Tempo ตามลำดับ ดังนี้

```
Spin1_SpinUp

If mciTempo% < 32767 Then
        mciTempo% = mciTempo% - 1
```

```
Spin1_SpinDown

If mciTempo% > 0 Then
    mciTempo% = mciTempo% + 1
```

ส่วนโค้ดที่เหลือก็เหมือนกันทุกประการ ซึ่งเป็นการกำหนดค่า Tempo ให้กับไฟล์ MIDI ดังนี้

```
        mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                      lpstrReturnString, lpstrReStrLen%, 0)

        mciErrCode& = mciSendString("status mciVBIS tempo", _
                      lpstrReturnString, lpstrReStrLen%, 0)
        mciTempo% = Val(lpstrReturnString)
        Text5.Text = mciTempo%
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
```

ส่วนในขณะที่กำลังเล่นกลับดนตรีนั้น โปรแกรมก็จะมีการเรียกใช้ Timer เพื่อแสดงผลตำแหน่งการเล่นกลับและสถานะในปัจจุบันของไดรเวอร์ในทุก ๆ ช่วงเวลา 1 วินาที ซึ่งก็จะมีการเรียกใช้รูทีนเหตุการณ์ Timer1_Timer() โดยที่โค้ดในรูทีนเหตุการณ์นี้ก็จะมีการทำงานใน 2 ส่วน ดังนี้

1. อ่านสถานะปัจจุบันของการเล่นกลับและทำการอัพเดตตำแหน่งของ scroll bar
2. ตรวจสอบสถานะการเล่นกลับ เพื่ออัพเดต Caption ของปุ่มคำสั่ง Play

```
mciErrCode& = mciSendString("status mciVBIS mode", lpstrReturnString, _
              lpstrReStrLen%, 0)
Text4.Text = Ztrim(lpstrReturnString)
If Text4.Text = "playing" Then
    Command1.Caption = "&Stop"
    ' Get current position (ms).
    mciErrCode& = mciSendString("status mciVBIS position", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    CurrentPosition& = Val(lpstrReturnString)
    Text4.Text = "playing " & CurrentPosition&
    HScroll1.Value = CInt(CurrentPosition& / 1000 + 1)
Else
    Command1.Caption = "&Play"
End If
```

สำหรับในตอนต่อไปผู้เขียนก็จะเขียนถึงวิธีการปรับแต่ง effect ต่าง ๆ ของไฟล์ .wav ด้วยฟังก์ชันของไฟล์ไลบรารี WAVETOOL.DLL ซึ่งเป็นไฟล์ไลบรารีที่ผู้เขียนได้พัฒนาขึ้นมาพิเศษสำหรับโปรแกรมเมอร์ VB/Win ได้นำไปใช้ในการจัดการกับไฟล์ .wav เช่น การปรับความดังเสียง (volume) การทำเสียงสะท้อน (echo) หรือการปรับความเร็ว (tempo) เป็นต้น ซึ่งการเขียนโค้ดที่จัดการกับเทคนิคเหล่านี้ด้วย VB/Win นั้น ค่อนข้างทำได้ยาก และที่สำคัญยังทำให้โปรแกรมที่ได้ทำงานช้าลงเป็นอย่างมาก ดังนั้นผู้เขียนจึงได้พัฒนาให้อยู่ในรูปแบบของไฟล์ไลบรารี .DLL เพื่อให้โค้ดที่ได้เป็นเนทีฟ (native code) จริง ๆ ซึ่งจะส่งผลให้สามารถทำงานได้รวดเร็วขึ้นมาก และจากการทดสอบของผู้เขียนปรากฏว่าสามารถประมวลผลได้เร็วกว่าโค้ดที่เขียนด้วย VB/Win ประมาณ 2-3 เท่า และที่สำคัญผู้เขียนยังอนุญาตให้ผู้อ่านสามารถนำไฟล์ไลบรารี WAVETOOL.DLL ไปใช้งานในเชิงธุรกิจได้ฟรีอีกด้วย พบกันใหม่ในฉบับหน้าสวัสดีครับ

ลิสติงที่ 1
ไฟล์โปรเจ็กต์
MIDIDEMO.VBP

```
Form=MIDIDEMO.FRM
Object={B16553C3-06DB-101B-85B2-0000C009BE81}#1.0#0; SPIN16.OCX
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.0#0; COMDLG16.OCX
ProjWinSize=126,0,248,215
ProjWinShow=2
IconForm="Form1"
HelpFile=""
Name="MIDIDEMO"
HelpContextID="0"
StartMode=0
VersionCompatible="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="VBIS"
```

ลิสติงที่ 2
ไฟล์ฟอร์ม
MIDIDEMO.FRM

```
VERSION 4.00
Begin VB.Form Form1
   Caption         =   "MIDI File Payer"
   ClientHeight    =   4800
   ClientLeft      =   1590
   ClientTop       =   1230
   ClientWidth     =   5880
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   1
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   5205
   Left            =   1530
   LinkTopic       =   "Form1"
   LockControls    =   -1  'True
   ScaleHeight     =   4800
   ScaleWidth      =   5880
   Top             =   885
   Width           =   6000
   Begin VB.CommandButton Command5
      Caption         =   "E&xit"
      Height          =   495
      Left            =   4680
```

→

ลิสติงที่ 2 (ต่อ)

```
    TabIndex        =   14
    Top             =   4200
    Width           =   1100
End
Begin VB.HScrollBar HScroll1
    Enabled         =   0   'False
    Height          =   240
    LargeChange     =   10
    Left            =   120
    TabIndex        =   10
    Top             =   4440
    Width           =   3135
End
Begin VB.CommandButton Command6
    Caption         =   "Select &File"
    Height          =   495
    Left            =   3360
    TabIndex        =   0
    Top             =   4200
    Width           =   1215
End
Begin VB.PictureBox Picture1
    Height          =   3975
    Left            =   120
    ScaleHeight     =   3945
    ScaleWidth      =   5625
    TabIndex        =   2
    Top             =   120
    Visible         =   0   'False
    Width           =   5655
    Begin VB.TextBox Text5
        Height          =   375
        Left            =   4320
        TabIndex        =   15
        Top             =   3480
        Width           =   855
    End
    Begin VB.CommandButton Command1
        Caption         =   "&Play"
        Height          =   495
        Left            =   120
        TabIndex        =   13
        Top             =   3360
        Width           =   1100
    End
    Begin VB.CommandButton Command2
        Caption         =   "Pa&use"
        Height          =   495
        Left            =   1245
        TabIndex        =   12
        Top             =   3360
        Width           =   1100
    End
    Begin VB.Timer Timer1
        Enabled         =   0   'False
        Interval        =   1000
        Left            =   120
        Top             =   120
    End
    Begin VB.TextBox Text4
        Height          =   285
        Left            =   4080
        TabIndex        =   6
        Top             =   3000
        Width           =   1455
    End
    Begin VB.TextBox Text3
        Height          =   285
        Left            =   1560
        TabIndex        =   9
        Top             =   3000
```

→

ลิสติงที่ 2 (ต่อ)

```
        Width           =   1575
End
Begin VB.TextBox Text2
    Height          =   495
    Left            =   1440
    MultiLine       =   -1  'True
    ScrollBars      =   2   'Vertical
    TabIndex        =   4
    Top             =   2400
    Width           =   4095
End
Begin VB.TextBox Text1
    Height          =   2175
    Left            =   120
    MultiLine       =   -1  'True
    ScrollBars      =   2   'Vertical
    TabIndex        =   1
    Top             =   120
    Width           =   5415
End
Begin Spin.SpinButton Spin1
    Height          =   375
    Left            =   5160
    TabIndex        =   16
    Top             =   3480
    Width           =   375
    _version        =   65536
    _extentx        =   661
    _extenty        =   661
    _stockprops     =   73
    backcolor       =   12632256
    tdthickness     =   1
End
Begin MSComDlg.CommonDialog CMDialog1
    Left            =   3960
    Top             =   0
    _version        =   65536
    _extentx        =   847
    _extenty        =   847
    _stockprops     =   0
    cancelerror     =   -1  'True
End
Begin VB.Label Label4
    Appearance      =   0   'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "Tempo"
    ForeColor       =   &H80000008&
    Height          =   255
    Left            =   3720
    TabIndex        =   7
    Top             =   3600
    Width           =   615
End
Begin VB.Label Label2
    Appearance      =   0   'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "Status"
    ForeColor       =   &H80000008&
    Height          =   255
    Left            =   3480
    TabIndex        =   8
    Top             =   3000
    Width           =   615
End
Begin VB.Label Label1
    Appearance      =   0   'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "Error message :"
    ForeColor       =   &H80000008&
    Height          =   255
    Index           =   0
```

→

ลิสติงที่ 2 (ต่อ)

```
            Left            =   120
            TabIndex        =   3
            Top             =   2520
            Width           =   1215
         End
         Begin VB.Label Label1
            Appearance      =   0  'Flat
            BackColor       =   &H00C0C0C0&
            Caption         =   "Return message :"
            ForeColor       =   &H80000008&
            Height          =   255
            Index           =   2
            Left            =   120
            TabIndex        =   5
            Top             =   3000
            Width           =   1455
         End
      End
      Begin VB.Label Label3
         Appearance      =   0  'Flat
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "MIDI Position (ms)"
         ForeColor       =   &H80000008&
         Height          =   195
         Left            =   120
         TabIndex        =   11
         Top             =   4200
         Width           =   1545
      End
   End
   Attribute VB_Name = "Form1"
   Attribute VB_Creatable = False
   Attribute VB_Exposed = False
   Option Explicit
   '
   ' Project name  : MIDIDEMO.VBP
   ' Program name  : MIDI File Palyer Program
   ' File name     : MIDIDEMO.FRM
   ' DLL filename  : MMSYSTEM.DLL
   ' Copyright (C)1996 by SUTTHISAK PHONGTHANAPANICH
   '

   '
   ' Windows API Multimedia functions & constants declaration.
   '
   Private Const WM_USER = &H400
   Private Const EM_SETREADONLY = (WM_USER + 31)
   Private Declare Function SendMessage Lib "User" (ByVal hWnd As Integer, _
           ByVal wMsg As Integer, ByVal wParam As Integer, lParam As Any) As Long
   Private Declare Function mciSendString Lib "mmsystem" _
           (ByVal lpstrCommand As String, ByVal lpstrReturnString As String, _
           ByVal uReturnlength As Integer, ByVal hWndCallback As Integer) As Long
   Private Declare Function mciGetErrorString Lib "mmsystem" (ByVal wError As Long, _
           ByVal lpstrBuffer As String, ByVal uLength As Integer) As Integer

   Dim mciFilename$, mciTempo%, mciMIDIlength&
   Dim Dummy%, OpenFlag As Boolean
   Dim mciErrCode&, mcilpstrReturnString As String * 256, mcilpstrReStrLen%
   Dim lpstrReturnString As String * 512, lpstrReStrLen%

   Private Sub Command1_Click()
       If Command1.Caption = "&Play" Then
           Command1.Caption = "&Stop"
           mciErrCode& = mciSendString("play mciVBIS from 1", _
                         lpstrReturnString, lpstrReStrLen%, 0)


           ' Call this procedure to set tempo.
           Text5_KeyPress 13
```

→

ลิสติงที่ 2 (ต่อ)

```
    Else
        Command1.Caption = "&Play"
        mciErrCode& = mciSendString("stop mciVBIS", _
                      lpstrReturnString, lpstrReStrLen%, 0)
    End If
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
End Sub

Private Sub Command2_Click()
    If Command2.Caption = "Pa&use" Then
        mciErrCode& = mciSendString("pause mciVBIS", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        Command2.Caption = "&Resume"
    Else
        mciErrCode& = mciSendString("resume mciVBIS", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        Command2.Caption = "Pa&use"
    End If
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
End Sub

Private Sub Command5_Click()
    Unload Me
End Sub

Private Sub Command6_Click()
    Dim p$

    On Error Resume Next

    CMDialog1.filename = mciFilename$
    CMDialog1.Filter = "MIDI Files|*.mid|All Files|*.*"
    CMDialog1.Flags = cdlOFNFileMustExist
    CMDialog1.ShowOpen

    If Err = 32755 Then Exit Sub

    mciFilename$ = CMDialog1.filename

    ' Close last opened media.
    If OpenFlag Then
        mciErrCode& = mciSendString("close mciVBIS", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
    OpenFlag = True
    ' Open media.
    mciErrCode& = mciSendString("open " & mciFilename$ & _
                  " type sequencer alias mciVBIS", lpstrReturnString, lpstrReStrLen%, 0)

    ' Set time format to millisec.
    mciErrCode& = mciSendString("set mciVBIS time format ms", _
                  lpstrReturnString, lpstrReStrLen%, 0)
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString

    ' Get information of current media (opened file).
    p$ = "- MIDI Media Information -" & vbCrLf & vbCrLf
    mciErrCode& = mciSendString("info mciVBIS file", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    p$ = p$ & "File name : " & Ztrim(lpstrReturnString) & vbCrLf
    mciErrCode& = mciSendString("info mciVBIS product", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    p$ = p$ & "Product : " & Ztrim(lpstrReturnString) & vbCrLf
    mciErrCode& = mciSendString("capability mciVBIS device type", _
                  lpstrReturnString, lpstrReStrLen%, 0)
    p$ = p$ & "Device Type : " & Ztrim(lpstrReturnString) & vbCrLf
```

→

ลิสติงที่ 2 (ต่อ)

```
    mciErrCode& = mciSendString("status mciVBIS division type", _
                  lpstrReturnString, lpstrReStrLen%, 0)
    p$ = p$ & "File Division Type : " & Ztrim(lpstrReturnString) & vbCrLf
    mciErrCode& = mciSendString("status mciVBIS length", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    mciMIDIlength& = Val(lpstrReturnString)
    p$ = p$ & "Song Length : " & mciMIDIlength& & vbCrLf
    mciErrCode& = mciSendString("status mciVBIS port", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    p$ = p$ & "MIDI Port : " & Ztrim(lpstrReturnString) & vbCrLf
    mciErrCode& = mciSendString("status mciVBIS time format", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    p$ = p$ & "Time Format : " & Ztrim(lpstrReturnString) & vbCrLf
    Text1.Text = p$

    ' Set wave length (ms) to HScroll.
    HScroll1.Min = 1
    HScroll1.Max = mciMIDIlength& / 1000

    ' Get default tempo.
    mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    mciTempo% = Val(lpstrReturnString)
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    Text5.Text = mciTempo%

    ' Show picture container if file selected.
    Picture1.Visible = True
    Timer1.Enabled = True
    HScroll1.Enabled = True
End Sub

Private Sub Form_Load()
    ' Set Text1 - Text4 to read only textbox.
    Dummy% = SendMessage(Text1.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text2.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text3.hWnd, EM_SETREADONLY, 1, ByVal 0&)
    Dummy% = SendMessage(Text4.hWnd, EM_SETREADONLY, 1, ByVal 0&)

    mcilpstrReStrLen% = Len(mcilpstrReturnString) - 1
    lpstrReStrLen% = Len(lpstrReturnString) - 1

    ' Show form at middle of screen.
    Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height - Me.Height) / 2
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    If OpenFlag Then
        mciErrCode& = mciSendString("close mciVBIS", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    End
End Sub

Private Sub HScroll1_Scroll()
    ' Disable Timer before seek.
    Timer1.Enabled = False

    mciErrCode& = mciSendString("seek mciVBIS to " & 1000& * HScroll1.Value, _
                  lpstrReturnString, lpstrReStrLen%, 0)
    mciErrCode& = mciSendString("play mciVBIS ", lpstrReturnString, lpstrReStrLen%,
0)
```

→

ลิสติงที่ 2 (ต่อ)

```
    ' Set tempo.
    mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                  lpstrReturnString, lpstrReStrLen%, 0)
    mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    mciTempo% = Val(lpstrReturnString)
    Text5.Text = mciTempo%
    Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, mcilpstrReStrLen%)
    ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString

    ' Enable Timer after play.
    Timer1.Enabled = True
End Sub

Private Sub ShowMCIreturnmessage(lpstrReturnString As String, mcilpstrReturnString As String)
    Text3.Text = Ztrim(lpstrReturnString)
    Text2.Text = Ztrim(mcilpstrReturnString)
End Sub

Private Sub Spin1_SpinDown()
    If mciTempo% > 0 Then
        mciTempo% = mciTempo% - 1
        mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                      lpstrReturnString, lpstrReStrLen%, 0)

        mciErrCode& = mciSendString("status mciVBIS tempo", _
                      lpstrReturnString, lpstrReStrLen%, 0)
        mciTempo% = Val(lpstrReturnString)
        Text5.Text = mciTempo%
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
End Sub

Private Sub Spin1_SpinUp()
    If mciTempo% < 32767 Then
        mciTempo% = mciTempo% + 1
        mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                      lpstrReturnString, lpstrReStrLen%, 0)

        mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        mciTempo% = Val(lpstrReturnString)
        Text5.Text = mciTempo%
        Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                 mcilpstrReStrLen%)
        ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
    End If
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
    Dim tmp

    Select Case KeyAscii
    Case 48 To 57: ' accept value.
    Case 13
        tmp = Val(Text5.Text)
        If tmp <= 32767 Then
            mciTempo% = tmp
            mciErrCode& = mciSendString("set mciVBIS tempo " & mciTempo%, _
                          lpstrReturnString, lpstrReStrLen%, 0)

            mciErrCode& = mciSendString("status mciVBIS tempo", lpstrReturnString, _
                          lpstrReStrLen%, 0)
            mciTempo% = Val(lpstrReturnString)
            Text5.Text = mciTempo%
            Dummy% = mciGetErrorString(mciErrCode&, mcilpstrReturnString, _
                     mcilpstrReStrLen%)
            ShowMCIreturnmessage lpstrReturnString, mcilpstrReturnString
        End If
```

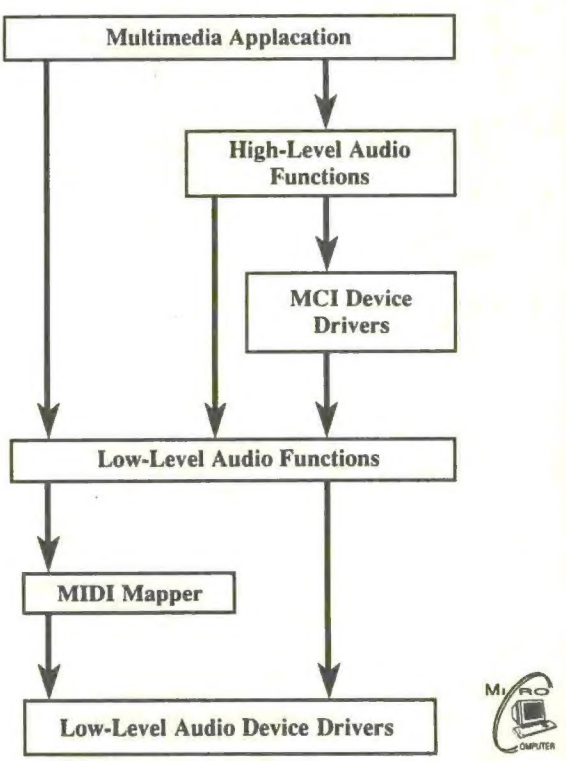
→

ลิสติงที่ 2 (ต่อ)

```
    End Select
End Sub

Private Sub Timer1_Timer()
    Dim CurrentPosition&

    On Error Resume Next

    mciErrCode& = mciSendString("status mciVBIS mode", lpstrReturnString, _
                  lpstrReStrLen%, 0)
    Text4.Text = Ztrim(lpstrReturnString)
    If Text4.Text = "playing" Then
        Command1.Caption = "&Stop"
        ' Get current position (ms).
        mciErrCode& = mciSendString("status mciVBIS position", lpstrReturnString, _
                      lpstrReStrLen%, 0)
        CurrentPosition& = Val(lpstrReturnString)
        Text4.Text = "playing " & CurrentPosition&
        HScroll1.Value = CInt(CurrentPosition& / 1000 + 1)
    Else
        Command1.Caption = "&Play"
    End If
End Sub

'
' Function : Remove ASCII 0 from end of string.
'
Private Function Ztrim(msg$) As String
    Dim i%, tmp$

    i% = InStr(msg$, vbNullChar)
    If i% Then
        tmp$ = RTrim$(Left$(msg$, i% - 1))
    Else
        tmp$ = RTrim$(msg$)
    End If
    Ztrim = tmp$
End Function
```

■

เทคนิคประจำคอลัมน์

ถาม สถาปัตยกรรมของวินโดวส์ด้านมัลติมีเดียนั้นมีความสัมพันธ์กันอย่างไร ?

ตอบ วินโดวส์ได้วางสถาปัตยกรรมด้านมัลติมีเดียไว้ 3 ระดับ ดังในรูป